APPENDIX A

SOFTWARE DESIGN DESCRIPTION AUTOMATIC NAME CLASSIFIER FOR CLASS-E (ANC-E)

TABLE OF CONTENTS

SOFTWARE DESIGN DESCRIPTION AUTOMATIC NAME CLASSIFIER FOR CLASS-E (ANC-E)

1. INTRODUCTION

1.1. Project Background

1.1.1. Legacy Consular Lookout And Support System (CLASS) and CLASS-E

The Consular Lookout and Support System (CLASS) performs namechecks of visa and passport applicants in support of the issuance process. Used by United States passport agencies, consulates, and border inspection agencies, CLASS serves as an automated index to manual files. CLASS is a centralized system residing on mainframe computers at the Department of State in Washington, DC. The Bureau of Consular Affairs, Consular Systems Division (CA/EX/CSD) of the Department of State (DOS) has responsibility for development, maintenance, and operation of CLASS.

CLASS was implemented in 1989; since that time, major advancements have occurred in database management systems, large-scale computers and their operating systems, and data telecommunications. In addition, name-matching techniques have also evolved based on the DOS's experience with the system and further linguistic research. This has led DOS in determining the necessity for a newer, more modernized system, CLASS-E (Consular Lookout and Support System-Enhanced).

The CLASS-E modernized version of automated name-matching will incorporate state-of-the-art hardware, data telecommunications, and database management technology to migrate the CLASS application from its Virtual Storage Access Method (VSAM) environment into a DB2 relational database system. In addition to providing virtually uninterrupted access to the lookout databases 24 hours a day, 7 days a week to the VO, PPT, overseas posts and support users, this enhanced system will position CLASS-E to incorporate advanced culturally-sensitive namecheck methods.

1.1.2. Culturally Sensitive Name Searching in CLASS-E

Personal naming systems vary widely from culture to culture. That is, names from around the world do not necessarily fit cleanly into the

Anglophone name model. Several of the manifestations of these differences are

· Anglicization of Non-English sound patterns (Mladevic written as Miladevich)
· Variant romanization schemes (Arabic Waseem ~ Ouassime, Shareef ~ Cherife; Chinese Xia ~ Hsia ~ Sya
· Dialectal variants (Arabic Abu Bakir [Egyptian] ~ Boubker [Moroccan]; Chinese Wu [Mandarin] ~ Ng [Cantonese, Fukien])
· Variant roman spelling conventions (French silent letters, German sch for English sh)

When dealing with Arabic and Chinese names and those of other languages that do not use the Roman alphabet, foe example, one quickly discovers one major source of name variation lies in how names are transliterated into roman characters from the original scripts. For both Arabic and Chinese, there are numerous competing transliteration standards, as well as less formal traditions. Xia, Hsia, and Sya, for example, are all romanized variants of the same Chinese name. Kassim, Qasim, Casem, Kacem and Asim are romanized variants of the same Arabic name. In Arabic, name variation often goes beyond the phonetic level. Analyzable elements such as "Abu" show up in many different forms, depending on dialect (e.g., Abu Bakir ~ Boubker). In Chinese, multiple traditions of transliteration are one of the sources of name variation; dialect issues also abound (e.g., Wu ~ Ng). Hispanic names, which make up the largest portion of the data base, place information value on name parts in a manner that is not consistent with Anglophone naming conventions. Exploitation of this culturally-specific information in the name search process leads to improved precision, recall, and overall system performance.

1.1.3. Automatic Name Classifier-E (ANC-E) in CLASS-E

The need for automatic name classification has become a become an undisputed necessary first step in the process of applying linguistic knowledge to solve the problems associated with name searching in large multicultural databases. In this environment, name classification serves as a means of routing queries to the proper language- and culture-specific algorithms. Currently, Legacy CLASS supports a single module, called ANI, which begins to address this need by returning a Boolean value indicating whether a name is or is not Arabic. If a name qualifies as Arab, it is subject to processing by an initial implementation of the Arabic algorithm designed by LAS for the State Department. Currently the expanding needs of the State Department are being addressed in the development a second culture-specific algorithm which will handle Hispanic names. The addition of a Hispanic algorithm to CLASS's

functionality requires the addition of a method for identifying Hispanic names in a manner similar that of ANI.

At this juncture, it is reasonable to turn enhancement efforts towards the development of a single, integrated, expandable algorithm for name classification which will address the need for classifying Arabic and Hispanic names, and which will anticipate the imminent addition of other languages. The integrated automatic name classification algorithm will represent a significant improvement over the existing ANI algorithm in that it will incorporate more linguistic knowledge, it will allow for future expansion with minimal coding effort, and it will allow information about a record's country of birth (COB) to contribute to the query routing decision. Figure 1-1 displays the integration of ANC-E within the CLASS-E system.

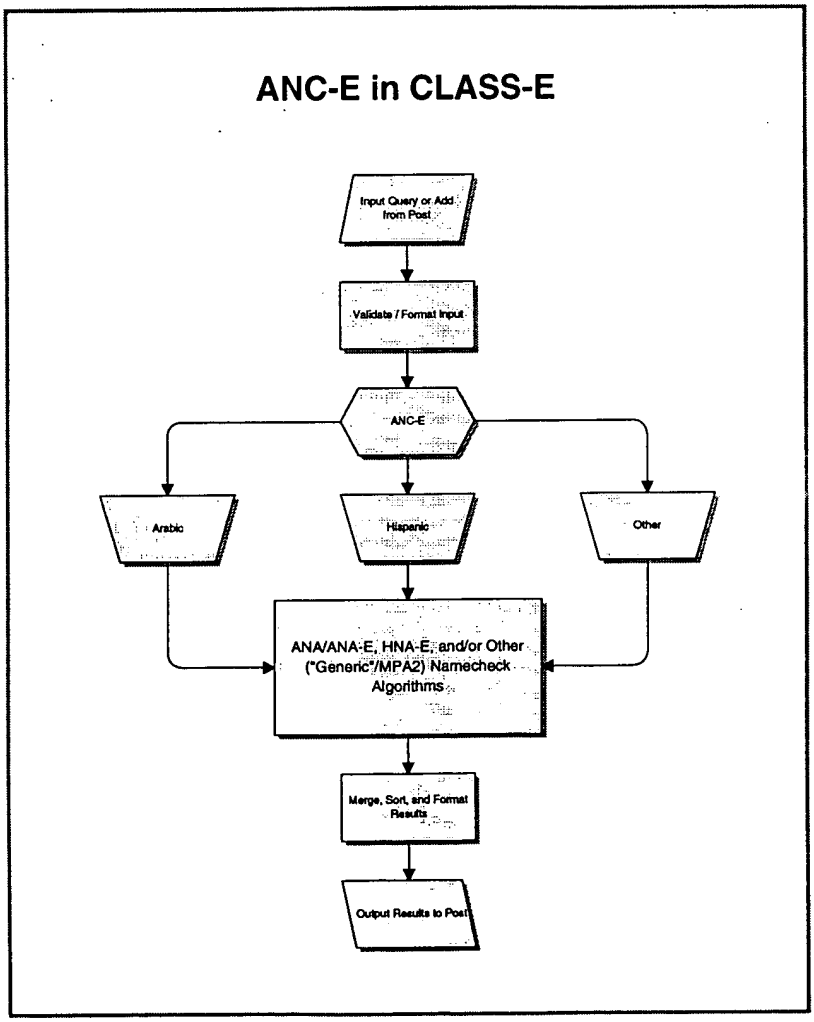


Figure 1-1

1.2. Scope

This document describes the linguistic motivation, requirements, and high level design for an Automatic Name Classifier (ANC) which will automatically determine whether a name qualifies as Hispanic or Arabic. The document's purpose is to provide information about the proposed design in order to facilitate the analysis and planning necessary to prepare for eventual implementation.

Intended to serve as the module that will provide for the integration of the enhanced Arabic Name Search Algorithm for CLASS-E (ANA-E) and the Hispanic Name Search Algorithm (HNA-E) into the overall CLASS-E architecture, the Automatic Name Classifier for CLASS-E (ANC-E) will provide the capability to automatically determine whether an input name is Arabic, Hispanic, or neither. In this system, names may be qualified as Arabic or Hispanic by virtue of passing one of two thresholds, or, conversely, may be disqualified as Arabic or Hispanic by virtue of having many characteristics of 'Other' types of names. The ANC-E system has been designed with an open architecture intended to facilitate the inclusion of additional cultures in the event that CLASS-E adds other culture-specific search algorithms in the future. Furthermore, since the ANC-E is data-driven, it is possible to tune its level of sensitivity for each individual culture being identified.

In CLASS-E the concept of the Legacy CLASS Multi-Pipe Architecture will be carried forward to include a distinct Arabic processing algorithm and a distinct Hispanic processing algorithm as well as perhaps others in the future. The type of processing to which an input name will be submitted will be a business decision of CA/EX/CSD and may to some degree be dependent on the impact that multiple processing of an input name would have on the performance of the system. It is likely that input names that are classified by the Advanced Name Classifier for CLASS-E (ANC-E) will be submitted to multiple of the following processors: the generic CLASS-E generic processing algorithm, the DOB processing algorithm, the ANA-E algorithm, and the HNA-E algorithm. The ANC-E will provide a determination as to which culture or cultures a name belongs; what use is made of this determination is a business decision of CA/EX/CSD. This decision will affect the design of the interface between the ANC-E and the rest of the CLASS-E system.

1.3. Definitions and Acronyms

1.3.1. Definitions

Affix* — A name *particle* which is neither a *title* nor a *qualifier*. Affixes in the ANC-E are defined as being delineated by white space; for example,

'de' in 'Tirso de Molina'. Note that, contrary to normal usage within linguistics, affixes are in contrast to (bound) *morphemes*, which are not delineated by white space.

Digraph — A two character *n-gram.*

Field — A data entry mechanism which allows the user to input a fixed number of characters. The fields typically referred to in the CLASS environment are the Given Name Field and the Surname Field.

Note that it is important to distinguish between *given name* and *surname* data entry Fields and *given name* and *surname* data elements, since data elements do not always occur in the proper field.

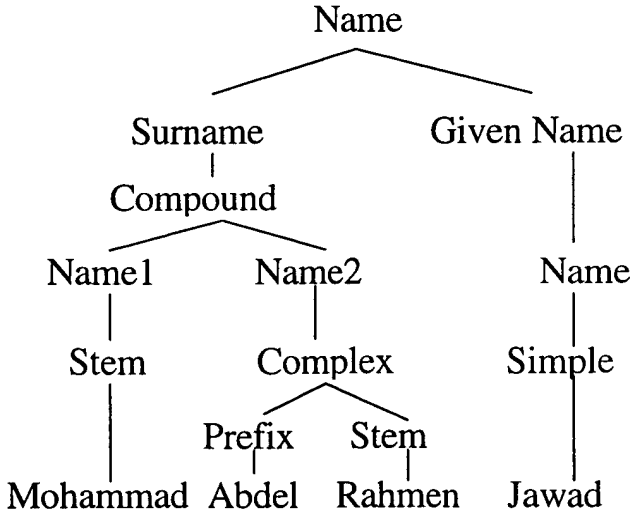
Given Name — The portion of a *name* which uniquely identifies an individual member of a family, as opposed to *surname.* Given Names may include one or more segments; for example, 'Mary Jane' in 'Mary Jane Cassoway'.

Infix — A substring occurring the middle of a name segment, but not at the edges. Both *n-grams* and *morphemes* may be infixes.

Morpheme* — (here, *bound* morpheme) A meaningful, variable length substring of a name segment. Morphemes may occur as *prefixes, infixes* or *suffixes.* Examples: '-ovitch' in 'Berkovitch'. Note that morphemes contrast with *affixes.*

Morphology — Referring to *morphemes.*

Name — The general term referring to the entire collection of *segments* which refer to a single person. A name may include one or more *given names,* one or more *surnames* and zero or more *particles.* For the purposes of ANC-E, a Name is considered to consist only of alphabetic characters and white space. The diagram below illustrates the relation of name parts to one another:



N-Gram — A variable length sequence of characters which serves as a useful

* Note that these terms have a slightly modified or restricted definition within the context of ANC-E.

	indicator of linguistic affinity, but which is not associated with a meaning. N-Grams may be considered to be indicators of the sound or spelling patterns of a language; for example, -ez is a Hispanic N-Gram.
Particle	A functional name element delineated by white space. *Titles*, *affixes* and *qualifiers* are the three kinds of particles identified in the ANC-E algorithm.
Prefix	A substring (*N-Gram* or *morpheme*) or a *particle* (*affix*) occurring at the beginning of a name segment.
Qualifier	A meaningful *particle* which represents a kinship relation or earned social status; for example, Jr. or Ph.D. Qualifiers typically occur at the end of a name field.
Segment	Any element within a name which is delineated by white space.
Suffix	A substring (*N-Gram* or *morpheme*) or a *particle* (*affix*) occurring at the end of a name segment.
Surname	The portion of a *name* which may indicate family membership, as opposed to *given name*. Surnames may include one or more segments and zero or more *particles*; for example, 'Fernandez de la Puente' in 'Hector Fernandez de la Puente'.
Syntax	The rules governing the order of name elements.
Title	A meaningful *particle* which represents a term of address and which typically occurs at the beginning of a name field. Examples: Dr. or Sir. Titles may be indicative of social position.
Trigraph	A three character *n-gram*.
Variant	An alternate spelling of a name *segment*; for example, Mohammad and Muhamed are variants of one another. Variants may be predictable, as in this example, or unpredictable, as evidenced by typographical or other data entry errors.

1.3.2. Acronyms

ANA	Legacy Arabic Namecheck Algorithm
ANA-E	Arabic Namecheck Algorithm for CLASS-E
ANC-E	Automatic Name Classifier for CLASS-E
ANI	Arabic Name Identification (of Legacy CLASS ANA)
ANR	Arabic Name Regularization (of Legacy CLASS ANA)
AOR	Application Owning Region
ARTP	Acceptance/Regression Test Plan
ARTR	Acceptance/Regression Test Report
BIMC	Beltsville Information Management Center
C/CE	CLASS to CLASS-E
CA	Bureau of Consular Affairs

CA/EX/CSD	Consular Affairs, Consular Systems Division
CAX	Consular Affairs Experimental (Development)
CCB	Configuration Control Board
CCR	Configuration Change Request
CDD	Critical Design Document
CDR	Critical Design Review
CE	CLASS-Enhanced
CICS	Customer Information Control System
CLASS	Consular Lookout and Support System
CLASS-E	Consular Lookout and Support System-Enhanced
CM	Configuration Management
CMOS	Complementary Metal Oxide Semiconductor
COB	Country of Birth
COR	Contracting Office Representative
CSD	Computer Systems Division
DBMS	Database Management System
DB2	IBM's relational database
DIA	Digraph Information Aggregator (of ANC-E)
DNC	Distributed Namecheck
DOB	Date of Birth
DOS	Department of State
FRR	Functional Requirements Review
FRS	Functional Requirements Specification
HNA-E	Hispanic Namecheck Algorithm for CLASS-E
IBIS	Interagency Border Inspection System
IDP1/IDP2	Intermediate Decision Processor 1 / 2 (of ANC-E)
IP	Installation Plan
IVV	Independent Verification and Validation
LIA	Linguistic Information Aggregator (of ANC-E)
LID	Linguistically Informed Decision Processor (of ANC-E)
LQA	Linguistic Quality Assurance
LQAR	Linguistic Quality Assurance Report
LSP	Linguistic Support Plan
LTF	Linguistic Trace Facility
NC	Namecheck
PC	Production Control
PMP	Project Management Plan
PPP	Post Phase-In Plan

PPT	Passport Office
PTS	Parallel Transaction Server
QA	Quality Assurance
QMF	Query Management Facility
QRP	Query Routing Processor
SA-1	State Annex-1
SESAP	Software Engineering Standards and Procedures
TAQ	Title, Affix Qualifier
TIR	Test Incident Report
TOR	Terminal Owning Region
TRR	Test Readiness Review
VO	Visa Office
VSAM	Virtual Storage Access Method

2. References

2.1. CLASS-E Project Management Plan (PMP)

2.2. CLASS-E Functional Requirements Specification (FRS)

2.2.1. Note: the CLASS-E FRS has not yet been finalized.

3. Decomposition Description

3.1. Module Decomposition

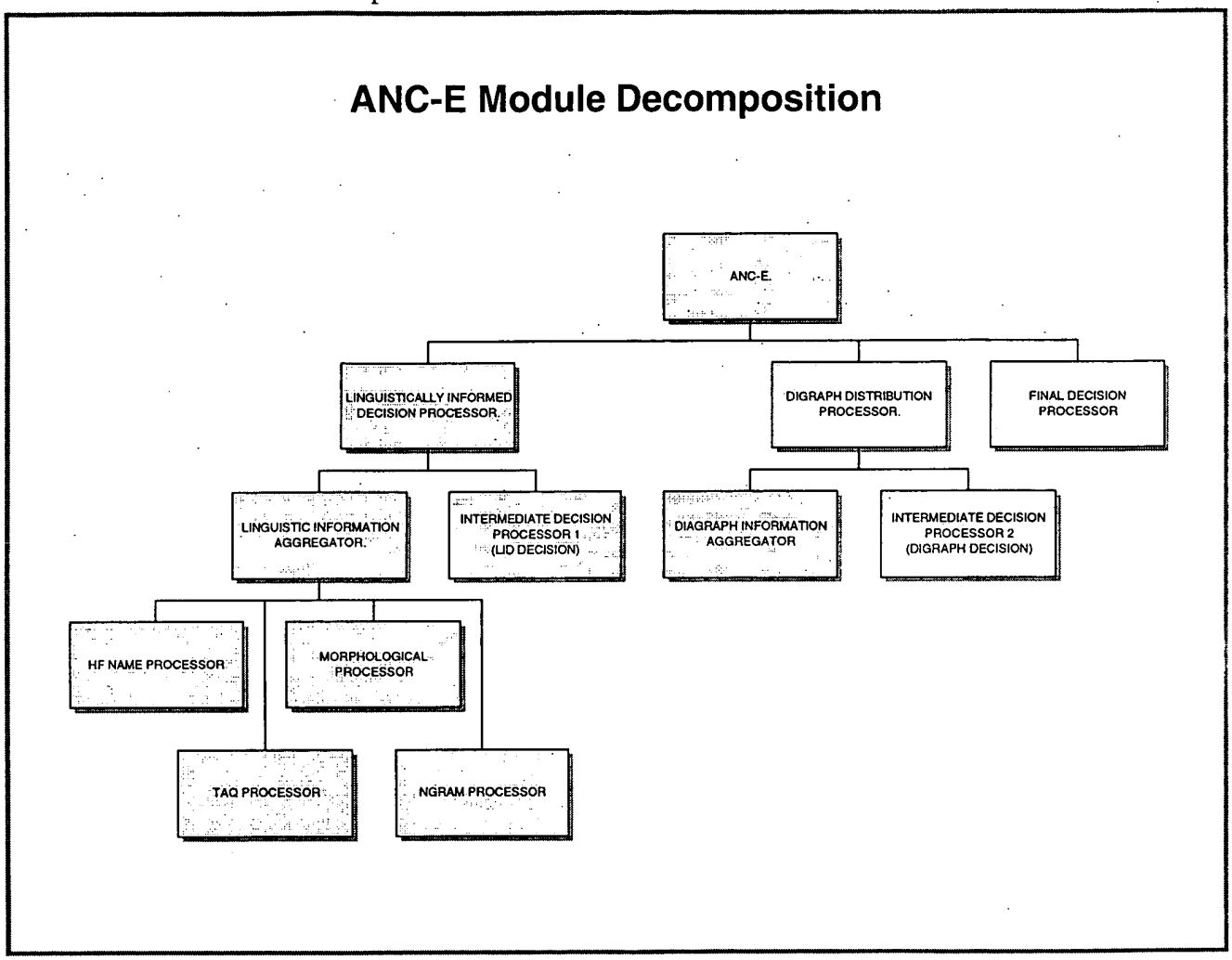


Figure 3-1

3.1.1. Automatic Name Classifier for CLASS-E (ANC-E) Module Decomposition

3.1.1.1. Identification

This program is referred to as the Automatic Name Classifier for CLASS-E (ANC-E).

3.1.1.2. Type

ANC-E is a program that is part of the larger CLASS-E system. It can be viewed as a "shell" program in that it is to

serve as a layer surrounding all of the culturally-specific name search algorithms implemented in CLASS-E.

3.1.1.3. Purpose

3.1.1.3.1. The need for automatic name classification is a necessary first step in the process of applying linguistic knowledge to solve the problems associated with name searching in large multicultural databases.

3.1.1.3.2. In the CLASS-E environment, name classification serves as a means of routing queries to the proper language- and culture-specific algorithms.

3.1.1.3.3. In addition to the rudimentary identification of Arabic names currently implemented in ANI, the addition of a Hispanic name search algorithm to CLASS-E's functionality requires the addition of a method for identifying Hispanic names.

3.1.1.3.4. ANC-E is a single, integrated algorithm for name classification which will address the need for classifying Arabic and Hispanic names, and which will anticipate the possible addition of other languages.

3.1.1.3.5. This integrated automatic name classification algorithm will represent a significant improvement over the existing ANI algorithm in that it will incorporate more linguistic knowledge, and will allow information about a record's country of birth (COB) to contribute to the query routing decision.

3.1.1.4. Function

3.1.1.4.1. The ANC-E will take as input a surname, given name, and COB in standard CLASS-E format.

3.1.1.4.1.1. There are two options with respect to the methodology for handling an input name and gathering the aggregate data

that will lead to the determination of cultural affinity for that name.

3.1.1.4.1.1.1. If ANC-E is to be implemented in an object-oriented environment, an object can be created which will contain all of the accumulated information to be used in the determination of cultural affinity. This object travels through the ANC-E system, thus allowing access to the accumulated information at any time. If ANC-E is integrated with the culturally-sensitive name search algorithms in CLASS-E, this option has the advantage that the all of the attendant linguistic information produced by ANC-E processing could be passed, along with the name, to the culturally-sensitive namecheck algorithm for further processing. That is, certain common linguistic processing would need to be performed only one time for the entire namecheck process, rather than once for each specific name search algorithm invoked.

3.1.1.4.1.1.2. If ANC-E is to be implemented in a non-object-oriented environment, ANC-E will process the name and COB as separate string values, and will output a either a single cultural affinity indicator (e.g. Arabic, Hispanic, or Other) or three Boolean values, one for each

culture under consideration, depending on the business decision made by CA/EX/CSD. If this option is chosen, linguistic processing information and scoring internal to ANC-E will not be available to outside processes.

3.1.1.4.2. The ANC-E will provide a determination as to which culture or cultures a name belongs.

3.1.1.4.3. The use that is made of the cultural affinity determinations made by ANC-E is a business decision of CA/EX/CSD (i.e. whether to allow a name to be processed by more than one namecheck algorithm, and whether ANC-E shall return more than one possible cultural affinity for a given input name). This decision will affect the design of the interface between the ANC-E and the rest of the CLASS-E system.

3.1.1.5. Subordinates

The following processes are subordinate to the main ANC-E program:

- The Linguistically Informed Decision Processor (LID)
- The Digraph Distribution Processor
- The Final Decision Processor.

3.1.2. Linguistically Informed Decision (LID) Module Decomposition

3.1.2.1. Identification

This module is referred to as the Linguistically Informed Decision Processor (LID).

3.1.2.2. Type

The LID is a module which contains two subordinate modules. The first subordinate module performs linguistic analysis, gathering linguistic information and scoring for the input name. The second subordinate module makes decisions as to the cultural affinity of the name, based on the scoring information gathered by the first module.

3.1.2.3. Purpose

3.1.2.3.1. The LID exists to provide a linguistically well-founded decision as to the cultural affinity of the input name.

3.1.2.3.2. As the first phase of processing, the LID addresses performance requirements by basing this decision on multiple readily observable linguistic factors, thus obviating the need for processing by the more intensive statistical digraph model and for reliance on name-external factors, such as Country of Birth (COB).

3.1.2.3.3. Furthermore, the LID provides a more linguistically-rich context in which to determine the cultural affinity of the input name than does it's purely digraph-distribution-based predecessor, ANI. Thus ANC-E is better able to identify names that are Hispanic or Arabic and to eliminate those that are not. Linguistic Indicators provide a rich source of information about the cultural affinity of a name. The LID processor will serve as a means of assuring that names which are strongly Arabic or Hispanic are qualified and, conversely, that names which have strong characteristics of some other culture are disqualified. Names which qualify as Hispanic, Arabic or 'Other' will not be submitted to the Digraph Analysis function.

3.1.2.4. Function

3.1.2.4.1. All linguistic indicator processing will take place before digraph analysis and will constitute a linguistically informed decision (LID) mechanism.

3.1.2.4.2. The LID accumulates and weighs factors from multiple knowledge sources in order to determine whether there is a sufficient amount of evidence to identify the input name as being Hispanic or Arabic, or, conversely, if there is enough

evidence to discount the possibility that the input name is either Hispanic or Arabic.

3.1.2.4.3. The LID will assign points to a name based on a weighted tabulation of scores from the following data sources:

- High Frequency name data
- TAQ data
- Morphological data
- Ngram data

3.1.2.4.4. The function of the LID is to determine a score for each cultural affinity being classified, and a score for 'Other'. For each culture, a name must get a score which passes its corresponding LID Threshold in order to be labeled as Arabic, Hispanic or "Other".

3.1.2.4.5. Each of the four types of linguistic indicator (listed in 3.1.2.4.3) will be associated with a set of four parameters, indicating the weight that a LID element is to be given.

3.1.2.4.6. The score for each language group will be calculated as a summation of the combination of the applicable factor times the score for each indicator found in the name string. Scoring details are included in the decomposition descriptions of the respective modules. (See sections 3.1.3 - 3.1.8.)

3.1.2.4.7. After all of the agents have processed the input name, the LID combines the detailed scoring information returned by the LIA to produce a LID score for Hispanic, Arabic, and for Other.

3.1.2.4.8. The LID passes the LID score to the Intermediate Decision Processor 1 for comparison to LID thresholds for cultures under consideration.

3.1.2.4.9. There are two alternatives for the output of the processing of the input name performed by the LID: an object containing linguistic processing information and scores or three Boolean values indicating whether the name has passed the LID

thresholds for Arabic, Hispanic, or Other. For more information, see 3.1.1.4.1.1.

3.1.2.4.10. If the LID identifies a name as Hispanic, Arabic, *or* Other (or any combination thereof), no further processing is required.

3.1.2.4.11. For a detailed example of LID processing, see the figures in Appendix A.

3.1.2.5. Subordinates

The following processes are subordinate to the LID:

- The Linguistic Information Aggregator (LIA)
- Intermediate Decision Processor 1 (LID Decision).

3.1.3. Linguistic Information Aggregator (LIA) Module Decomposition

3.1.3.1. Identification

This module is referred to as the Linguistic Information Aggregator (LIA).

3.1.3.2. Type

LIA is a module which contains four subordinate functions (agents) all of which contribute to the final decision or decisions made by the LID as to the cultural affinity of the input name. Thus, conceptually, LIA and the LID can be viewed as parts of a blackboard (voting) system, an expert system, or as parts of a system with multiple intelligent agents.

3.1.3.3. Purpose

3.1.3.3.1. The LIA exists to enable the linguistic decision made by the LID. The LIA controls the flow of information from the four linguistic agents subordinate to it.

3.1.3.3.2. If the implementation choices accompanying the object-oriented description of ANC-E are chosen (see 3.1.1.4.1.1.1), LIA could help performance by allowing certain linguistic processing to occur only once for each name check, rather than once for each algorithm invoked. (Note: In Legacy CLASS each algorithm is referred to as a separate 'pipe'.)

3.1.3.4. Function

3.1.3.4.1. LIA accumulates linguistic information factors from multiple knowledge sources for each culture under consideration (i.e. currently Hispanic, Arabic, and Other).

3.1.3.4.2. In cases of conflict, the order of precedence for identifying items within an input name is TAQ particle, Morpheme, Ngram.

3.1.3.4.2.1. If a string of letters is identified as a TAQ particle for a particular culture, a substring of that same string (including the entire string itself) cannot also be identified as a Morpheme or an Ngram for that same culture.

3.1.3.4.2.2. If a string is identified as a Morpheme for a particular culture, the characters that make up that Morpheme cannot also be considered as part of an Ngram for that culture.

3.1.3.4.2.3. HF Names from a given culture can contain Morphemes and / or Ngrams for that same culture; however, the precedence rules in sections 3.1.3.4.2.1 and 3.1.3.4.2.2 apply.

3.1.3.4.3. As the subordinate functions (agents) process the input name, detailed scoring information is collected by LIA, and weighted according to its information value as indicated in the LID Parameter data store.

3.1.3.4.4. After all of the agents have provided their input, the LIA returns this detailed scoring information to the LID.

3.1.3.4.5. For a detailed example of aggregation of information by LIA, see the figures in Appendix A.

3.1.3.5. Subordinates

The following processes are subordinate to the LIA:

- The High Frequency (HF) Name Processor
- The Title, Affix, Qualifier (TAQ) Processor
- The Morphological Processor
- The Ngram Processor.

3.1.4. High Frequency (HF) Name Processor Module Decomposition

3.1.4.1. Identification

This function is referred to as HF Name Processor.

3.1.4.2. Type

The HF Name Processor is a function which is invoked by the Linguistic Information Aggregator (LIA).

3.1.4.3. Purpose

Certain given names and surnames occur much more frequently in some cultures than in others. The name "Mohammed", for example occurs frequently in Arabic names. The surname "Rodriguez" lends support to the possibility that the name in question is Hispanic. The name "Nganga" in any position suggests that the name might not be either Arabic or Hispanic. The HF Name Processor exists to take advantage of the information available in high frequency names in the cultural identification of the name.

3.1.4.4. Function

3.1.4.4.1. For each name segment present in the input name, the HF Name Processor determines whether that name is present in the HF Name data store.

3.1.4.4.2. If the name is present in the HF name data store, the HF Name Processor retrieves and records the culture, name field (given name or surname), and score associated with that name from the data store.

3.1.4.4.3. Also recorded for each HF name found is whether it was found in position or out of position. For example, since "Rodriguez" is listed as a surname in the HF Names data store, if it is found in the GN field in the input name, it

is reported as a surname considered to be out of position.

3.1.4.4.4. The HF Name Processor tracks scoring information for each HF name found, and returns this detailed scoring information to LIA.

3.1.4.5. Subordinates

None.

3.1.5. Title, Affix, Qualifier (TAQ) Processor Module Decomposition

3.1.5.1. Identification

This function is referred to as the TAQ Processor.

3.1.5.2. Type

The TAQ Processor is a function which is invoked by the Linguistic Information Aggregator (LIA).

3.1.5.3. Purpose

As noted in section 1.3.1, name fields have a syntactic structure which may be simple, compound, complex, or compound-complex. Name fields which are complex or compound-complex contain particles: titles, affixes, or qualifiers. These particles can be used to further narrow the range of possibilities for the cultural affinity of the input name. The TAQ Processor exists to make use of the information available in particles.

3.1.5.4. Function

3.1.5.4.1. For each segment present in the input name, the TAQ Processor determines whether that segment is a particle present in the TAQ data store.

3.1.5.4.2. If the segment is present in the TAQ data store, the TAQ Processor retrieves and records the culture, name field (given name or surname), and score associated with that TAQ particle from the data store.

3.1.5.4.3. Also recorded for each TAQ particle found is whether it was found in position or out of position. (See example in section 3.1.4.4.3.)

3.1.5.4.4. The TAQ Processor tracks scoring information for each HF TAQ particle found, and returns this detailed scoring information to LIA.

3.1.5.5. Subordinates

None.

3.1.6. Morphological Processor Module Decomposition

3.1.6.1. Identification

This function is referred to as Morphological Processor.

3.1.6.2. Type

The Morphological Processor is a function which is invoked by the Linguistic Information Aggregator (LIA).

3.1.6.3. Purpose

As noted and defined in section 1.3.1, morphological elements, such as -ovich, can play a large part in determining the cultural affinity of an input name. The Morphological Processor exists to take advantage of this information in the name classification process.

3.1.6.4. Function

3.1.6.4.1. For each Morpheme present in the Morphology data store, the Morphological Processor determines whether that Morpheme is present in the input name.

3.1.6.4.1.1. Note that the above processing differs from that in the HF Name Processor (3.1.4.4) and the TAQ Processor (3.1.5.4). Since the Morphology data store contains only bound Morphemes, that is Morphemes not surrounded by white space, it is not possible to locate them based on name segments, which are surrounded by white space. Rather, it is necessary to determine if any of the items listed in the

Morphology data store is a substring of any of the name segments present in the input name, within certain constraints. For more detailed information on identifying Morphemes in the input name, see sections 3.2.4 (Morphological Data Store Data Decomposition) and 3.1.6 (Morphological Processor Module Decomposition).

3.1.6.4.2. For each Morpheme found in the input name, the Morphological Processor retrieves and records the morpheme found, the culture, name field (given name or surname), and score associated with that Morpheme from the data store.

3.1.6.4.3. Also recorded for each Morpheme found is whether it was found in position or out of position. (See example in section 3.1.4.4.3.)

3.1.6.4.4. The Morphological Processor tracks scoring information for each Morpheme found, and returns this detailed scoring information to LIA.

3.1.6.5. Subordinates

None.

3.1.7. Ngram Processor Module Decomposition

3.1.7.1. Identification

This function is referred to as the Ngram Processor.

3.1.7.2. Type

The Ngram Processor is a function which is invoked by the Linguistic Information Aggregator (LIA).

3.1.7.3. Purpose

As described in section 1.3.1, Ngrams are strings of letters that occur with statistical significance in names with a given cultural affinity. The Ngram Processor exists to take advantage of this statistical phenomenon in the name typing process.

3.1.7.4. Function

3.1.7.4.1. For each Ngram present in the Ngram data store, the Ngram Processor determines whether that Ngram is present in the input name.

3.1.7.4.1.1. Note that the above processing is similar to that in the Morphological Processor. (See section 3.1.6.4, and especially section 3.1.6.4.1.1 for a detailed note.)

3.1.7.4.2. For each Ngram found in the input name, the Ngram Processor retrieves and records the Ngram found, the culture, name field (given name or surname), and score associated with that Ngram from the data store.

3.1.7.4.3. Also recorded for each Ngram found is whether it was found in position or out of position. (See example in section 3.1.4.4.3.)

3.1.7.4.4. The Ngram Processor tracks scoring information for each Ngram found, and returns this detailed scoring information to LIA.

3.1.7.5. Subordinates

None.

3.1.8. Intermediate Decision Processor 1 (LID Decision) Module Decomposition

3.1.8.1. Identification

This module is referred to as Intermediate Decision Processor 1 (IDP1).

3.1.8.2. Type

IDP1 is a function which is invoked directly by the Linguistically Informed Decision Processor (LID).

3.1.8.3. Purpose

IDP1 is the decision-making function of the LID. It determines whether enough linguistic information has been gathered from the various intelligent agents by LIA to

confidently determine that the input name belongs to one of the cultures being identified (currently Arabic, Hispanic, and Other).

3.1.8.4. Function

3.1.8.4.1. IDP1 accepts as input one aggregate LID score for each culture being identified as well as an aggregate LID score for Other.

3.1.8.4.2. For each LID score, IDP1 compares that score to the LID threshold for the appropriate culture (or Other).

3.1.8.4.3. If the LID score is greater than or equal to the appropriate LID threshold, IDP1 returns a value of True for the culture in question. If the LID score is less than the LID threshold for the culture in question, IDP1 returns a value of False for the culture in question.

3.1.8.4.3.1. A True value indicates to the LID that enough evidence has been accumulated by LIA to confidently identify the name as belonging to the culture in question.

3.1.8.4.3.2. A False value indicates to the LID that not enough evidence has been accumulated by LIA to confidently identify the name as belonging to the culture in question.

3.1.8.4.3.3. A value of True can be returned for more than one cultural affinity.

3.1.8.4.3.4. A value of False may be returned for all cultural affinities.

3.1.8.4.4. Alternatively, IDP1 could return a value for each culture equal to the LID score minus the LID threshold for that culture.

3.1.8.4.4.1. Given the alternative above, the LID would interpret negative scores as

False values and nonnegative scores as True values.

3.1.8.4.4.2. The utility of this alternative is that if an object-oriented implementation is chosen, the values calculated by IDP1 could be incorporated into the object mentioned in section 3.1.1.4.1.1.1, and would be available as part of the information that the name object "knows" about itself for use in later processing.

3.1.8.4.5. If a return value of True for any culture (or for "Other") is obtained from IDP1, no further processing is required.

3.1.8.5. Subordinates

None.

3.1.9. Digraph Distribution Processor Module Decomposition

3.1.9.1. Identification

This module is referred to as the Digraph Distribution Processor.

3.1.9.2. Type

The Digraph Distribution Processor is a module which has two subordinate functions.

3.1.9.3. Purpose

The Arabic Name Identification (ANI) subprogram currently in use in Legacy CLASS is based purely on a model of digraph distribution in Arabic names. Digraph distribution information has proved useful in determining the cultural affinity of names. Based on a statistical model generated from digraph distribution statistics and initial and final trigraph statistics, the Digraph Distribution Processor lends additional information to the attempt to identify the provenance of the input name.

3.1.9.4. Function

3.1.9.4.1. The Digraph Distribution Processor takes as input the surname from the name input to

ANC-E. This portion of ANC-E operates only on surname data.

3.1.9.4.2. The Digraph Distribution Processor is invoked only when the LID has not been successful in assigning any cultural affinity to the input name. (See sections 3.1.2.4.10 and 3.1.8.4.5.)

3.1.9.4.3. The Digraph Distribution Processor calculates scores based on digraph distribution statistics for each culture in order to determine whether there is a sufficient amount of evidence to identify the input name as being Hispanic or Arabic. Note that there is no Digraph Distribution Score computed for Other.

3.1.9.4.4. The Digraph Distribution Parameters data store contains a Digraph Skew Factor for each cultural affinity.

3.1.9.4.5. The Total Digraph Distribution Score for the input name is equal to the Raw Digraph Distribution Score returned by the DIA plus the value of the Digraph Skew Factor for the appropriate culture.

3.1.9.4.6. The Digraph Distribution Processor passes the Total Digraph Distribution score for each culture to the Intermediate Decision Processor 2 for comparison to Digraph thresholds for cultures under consideration.

3.1.9.4.7. There are two alternatives for the output of the processing of the input name performed by the Digraph Distribution Processor: an object containing a Digraph Distribution Score for each culture, or two Boolean values indicating whether the name has passed the Digraph thresholds for Arabic or Hispanic. For more information, see 3.1.1.4.1.1.

3.1.9.4.8. If the Digraph Distribution Processor identifies a name as Hispanic, Arabic, or both no further processing is required.

3.1.9.5. Subordinates

The following processes are subordinate to the Digraph Distribution Processor:

- The Digraph Information Aggregator (DIA)
- Intermediate Decision Processor 2 (Digraph Decision).

3.1.10. Digraph Information Aggregator (DIA) Module Decomposition

3.1.10.1. Identification

This module is referred to as the Digraph Information Aggregator (DIA).

3.1.10.2. Type

The DIA is a process invoked by the Digraph Distribution Processor. The DIA operates only on surname segments consisting solely of alphabetic characters.

3.1.10.3. Purpose

The DIA gathers the information necessary for the Digraph Distribution Processor to determine whether there is sufficient information to identify the input name as Hispanic or Arabic.

3.1.10.4. Function

3.1.10.4.1. For purposes of DIA processing, a surname segment is defined as any string of characters delimited by white space.

3.1.10.4.1.1. Given a surname containing more than one part as input, the name is segmented (based on white space). Each part of multi-part surnames is processed separately, and the scores are combined in the manner described below.

3.1.10.4.2. DIA will calculate a score for each surname segment by totaling the scores for all digraphs within the surname segment.

3.1.10.4.2.1. The set of digraphs for a surname consists of all possible substrings of

two contiguous characters in the surname.

3.1.10.4.2.2. Word-boundaries are considered characters, so the additional digraphs "*word-boundary+first-letter*" and "*last-letter+word-boundary*" are included in the set of digraphs for each name.

3.1.10.4.2.3. In general, a surname segment of length n contains $(n+1)$ digraphs, ordered from leftmost to rightmost.

3.1.10.4.3. Each digraph in the surname segment is looked up in a table containing scores for all possible digraphs for all cultural affinities being scored. DIA maintains a cumulative total of all digraph scores assigned to a surname segment.

3.1.10.4.4. Likewise, scores are assigned for the initial and final trigraphs of each name segment.

3.1.10.4.5. The initial and final trigraph scores are added to the cumulative score for that segment. A score is thus calculated for each segment of the surname.

3.1.10.4.6. The Raw Digraph Distribution Score for the input name is equal to the sum of all individual surname segment scores thus calculated.

3.1.10.5. Subordinates

None.

3.1.11. Intermediate Decision Processor 2 (Digraph Decision) Module Decomposition

3.1.11.1. Identification

This module is referred to as Intermediate Decision Processor 2 (IDP2).

3.1.11.2. Type

IDP2 is a function which is invoked directly by the Digraph Distribution Processor.

3.1.11.3. Purpose

IDP2 is the decision-making function of the Digraph Distribution Processor. It determines whether enough digraph distribution information is present to confidently determine that the input name belongs to one of the cultures being identified (currently Arabic or Hispanic).

3.1.11.4. Function

3.1.11.4.1. IDP2 accepts as input one Digraph Distribution Score for each culture being identified.

3.1.11.4.2. For each Digraph Distribution Score, IDP2 compares that score to the Digraph threshold for the appropriate culture.

3.1.11.4.3. If the Digraph Distribution Score is greater than or equal to the appropriate Digraph threshold, IDP2 returns a value of True for the culture in question. If the Digraph Distribution Score is less than the Digraph threshold for the culture in question, IDP2 returns a value of False for the culture in question.

3.1.11.4.3.1.A True value indicates to the Digraph Distribution Processor that digraph distribution information is conclusive enough to confidently identify the name as belonging to the culture in question.

3.1.11.4.3.2.A False value indicates to the Digraph Distribution Processor that digraph distribution information is not conclusive enough to confidently identify the name as belonging to the culture in question.

3.1.11.4.3.3.A value of True can be returned for more than one cultural affinity.

3.1.11.4.3.4.A value of False may be returned for all cultural affinities.

3.1.11.4.4. Alternatively, IDP2 could return a value for each culture equal to the Digraph Distribution

Score minus the Digraph threshold for that culture.

3.1.11.4.4.1. Given the alternative above, the Digraph Distribution Processor would interpret negative scores as False values and nonnegative scores as True values.

3.1.11.4.4.2. The utility of this alternative is that if an object-oriented implementation is chosen, the values calculated by IDP2 could be incorporated into the object mentioned in section 3.1.1.4.1.1.1, and would be available as part of the information that the name object "knows" about itself for use in later processing.

3.1.11.5. Subordinates

None.

3.1.12. Final Decision Processor Module Decomposition

3.1.12.1. Identification

This module is referred to as the Final Decision Processor.

3.1.12.2. Type

The Final Decision Processor is a module invoked directly by the ANC-E main program.

3.1.12.3. Purpose

3.1.12.3.1. Although the LID and the Digraph Distribution Processor are each powerful methods for identifying the cultural affinity of names in themselves, some benefit can be gained from combining the judgments of these two modules when neither has been successful in reaching a conclusion within a reasonable level of certainty on its own.

3.1.12.3.2. Additionally, within the CLASS-E system, information about the Country of Birth (COB)

will usually be available. Although this information is not generally sufficient to determine the cultural affinity of a name in itself, it could provide the additional evidence necessary to reach a conclusion when combined with the judgments of the LID and the Digraph Distribution Processor.

3.1.12.3.3. The final decision processor exists to take all of this information into account, in an effort to determine the cultural affinity of the input name by combining all available data when the individual data elements themselves are not strong enough indicators.

3.1.12.4. Function

3.1.12.4.1. In the event that neither the LID nor the Digraph Distribution Processor is successful in determining a cultural affinity for the input name, the processing continues to the Final Decision Processor. (See sections 3.1.2.4.10, 3.1.8.4.5, and 3.1.9.4.8.)

3.1.12.4.2. If the options suggested in sections 3.1.1.4.1.1.1, 3.1.8.4.4, and 3.1.11.4.4 are incorporated into the implementation, the final Decision Processor will have access to all of the information it needs to perform its task encapsulated in the name information object. Otherwise, the Final Decision Processor will take as input LID scores (for each cultural affinity and for Other) and digraph scores (for each cultural affinity) for the input name.

3.1.12.4.3. For each culture still under consideration, the final decision processor will determine if the Digraph Distribution score for that culture is within the range specified by the Under_Di_Threshold parameter[1]. Note that since there is no Digraph Distribution score calculated for the cultural affinity "Other",

[1] For additional information regarding the range specified by the Under_Di_Threshold parameter, see section 3.2.9.4.2.5.

there is no Under_Di_Threshold parameter associated with Other, and this processing applies only to cultures included in the current name classifier (e.g. Arabic and Hispanic).

3.1.12.4.3.1. In the event that the Digraph Distribution score is in the range specified for the particular culture, processing continues to determine if there is enough additional evidence to identify the input name as belonging to that culture.

3.1.12.4.3.2. In the event that the Digraph Distribution score is not in the range specified for the particular culture, that cultural affinity is removed from further consideration for the input name.

3.1.12.4.4. For each culture still under consideration, the final decision processor will determine if the LID score for that culture is within the range specified by the Under_LID_Threshold parameter[1].

3.1.12.4.4.1. In the event that the LID score is in the range specified for the particular culture, the final decision processor will identify the input name as belonging to that culture.

3.1.12.4.4.2. In the event that the LID score is not in the range specified for the particular culture, processing continues to determine if there is enough additional evidence to identify the input name as belonging to that culture.

3.1.12.4.5. For each culture still under consideration, the Final Decision Processor determines whether

[1] For more information regarding the range specified by the Under_LID_Threshold parameter, see section 3.2.9.4.2.4.

the COB supplied with the input name is in the partition associated with the cultural affinity, as defined in the COB Proximity (COBPROX) Data Store.

3.1.12.4.5.1. In the event that the COB supplied with the input name is in the partition associated with the cultural affinity under consideration, the final decision processor will identify the input name as belonging to that culture.

3.1.12.4.5.2. In the event that the COB supplied with the query is not in the partition associated with the cultural affinity under consideration, that cultural affinity is removed from further consideration for the input name.

3.1.12.4.5.3. In the event that the COB supplied with the input name is Unknown (i.e. "XXX" in Legacy CLASS), the Final Decision Processor will identify the input name as belonging to the cultural affinity under consideration. Note that this is a conscious decision to err on the side of recall in the absence of adequate information (that is, to identify a name as belonging to a culture, perhaps erroneously, in an effort to avoid erroneously not identifying some input names as belonging to that culture). This is related to the other policy decisions to be made by CA, and may change based on those decisions.

3.1.12.4.6. A summarization of the processing performed by the final decision processor is contained in Figure 3-2.

```
IF (Di_Threshold - Digraph_Distribution_Score - Under_Di_Threshold >= 0) AND
   ((LID_Threshold - LID_Score - Under_LID_Threshold >= 0)                OR
   (COB_of_input_name is in partition OR COB_of_input _name is Unknown))
THEN
    Identify Input Name as belonging to the culture in question
END IF
```

Figure 3-2

3.1.12.5. Subordinates

None.

3.2. Data Decomposition

The data tables which underlie the Linguistically Informed Decision processor are crucial to the success of the algorithm. As discussed in 3.1.2.4.3, the linguistic data to be used are: High Frequency names, TAQ elements, Morphological elements and Ngrams. The entries for each of these linguistic sources will be associated, minimally, with a name field, a cultural group and a score. Also associated with the LID are control parameters. The Data entities accessed by the LID, as well as by other ANC-E Modules are depicted in Figure 3-3. This section describes in detail the data stores used by ANC-E. For examples of the type of information to be included in the data stores, see the detailed example in Appendix A.

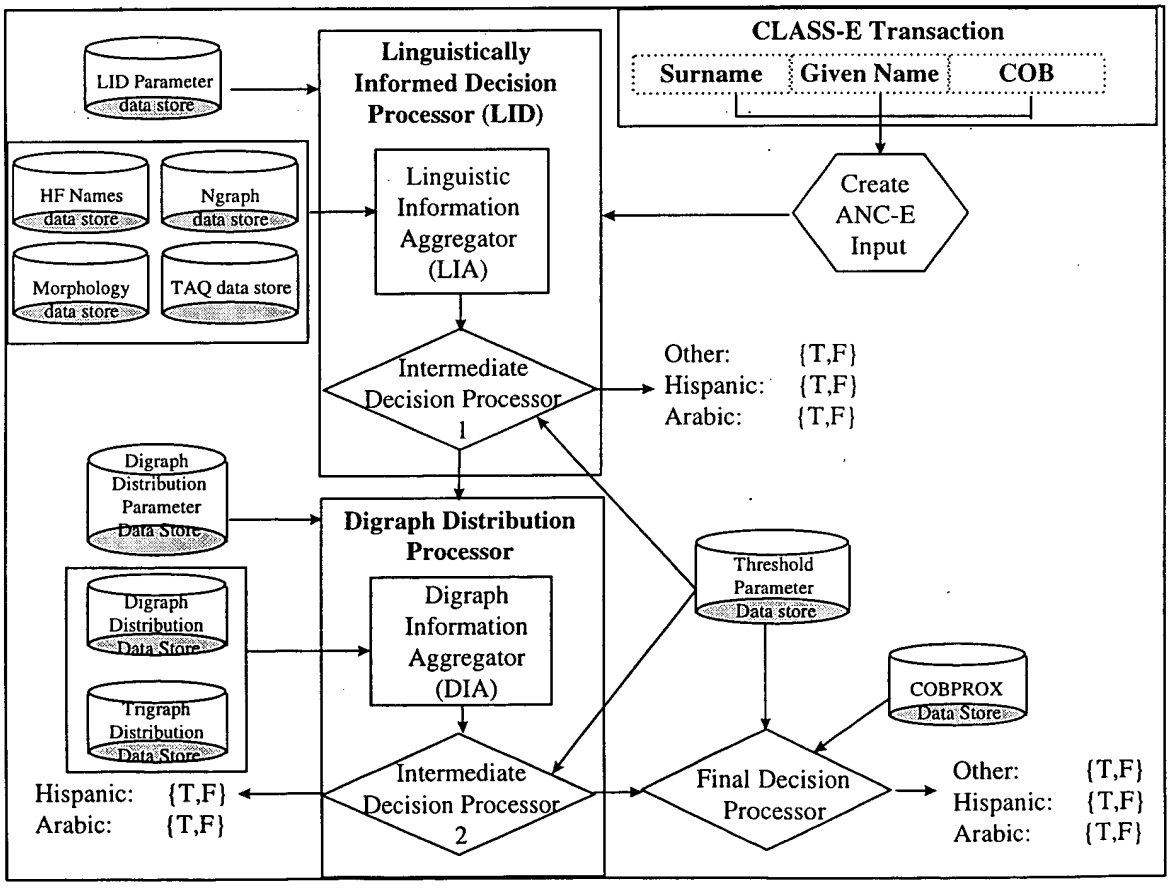


Figure 3-3

3.2.1. LID Parameter Data Store Data Decomposition

3.2.1.1. Identification

This data store is referred to as the LID Parameter Data Store.

3.2.1.2. Type

The LID Parameter Data Store is a data store that is accessed by the LID module.

3.2.1.3. Purpose

3.2.1.3.1. The LID takes factors such as HF names, Ngrams, TAQ particles, and Morphemes into account in determining the cultural affinity of the input name.

3.2.1.3.2. Although each of these factors is valuable, they should not all be given the same relative weight in determining the cultural affinity score of the input name.

3.2.1.3.3. Furthermore, as in all real-world applications, the data in the CLASS-E database is not "clean". That is, data elements are not always found in the expected positions. Therefore, it is common to find surname elements in the given name field, and vice versa. Since it is not always possible to determine whether a particular instance of "out-of-field" data is due to random factors influencing data entry procedures or to a name's being from a culture other than the one hypothesized, data found "out of position" should not be given as great a weight as data found in the canonical position.

3.2.1.3.4. The LID Parameter data store exists in order to allow for different weighting of evidence found by the LID based on the above factors without hard-coding the exact weighting scheme itself in the LID. This will allow for runtime fine-tuning and adjustments to ANC-E without the necessity of recompiling LID module code.

3.2.1.4. Function

3.2.1.4.1. Table 3.2-1 contains a description of the data to be contained in the LID Parameter data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
AGENT_NAME	character	10	{HFNAME, TAQ, MORPHOLOGY, NGRAM}
NAMEFIELD	character	1	{G,S}
INFIELD_SCORE	integer	2	{1,2, ..., 10}
OUT_OF_FIELD_SCORE	integer	2	{1, 2, ..., 10}

Table 3.2-1

3.2.1.4.2. The LID uses the information provided in this data store when calculating aggregate cultural affinity scores from the detailed scoring information returned by LIA.

3.2.1.4.2.1. AGENT_NAME indicates to which agent (function) the given INFIELD_SCORE and OUT_OF_FIELD_SCORE weightings apply.

3.2.1.4.2.2. NAMEFIELD indicates whether the INFIELD_SCORE and OUT_OF_FIELD_SCORE weightings apply to the Given Name (G) or to the Surname (S).

3.2.1.4.2.3. IN_FIELD_SCORE is the weighting to be applied to data elements' raw scores returned by the specified agent when found in the specified name field.

3.2.1.4.2.4. OUT_OF_FIELD_SCORE is the weighting to be applied to data elements' raw scores returned by the specified agent when found out of the specified name field.

3.2.1.4.2.4.1. For more information concerning IN_FIELD and OUT_OF_FIELD attributes returned from individual agents via LIA, see 3.1.4.4.3.

3.2.1.4.2.4.2. For an example of scoring of an input name using raw scores returned by agents and the LID Parameters, see Figure 3-4.

3.2.1.5. Subordinates

None.

3.2.2. High Frequency Name Data Store Data Decomposition

3.2.2.1. Identification

This data store is referred to as the HF Name Data Store.

3.2.2.2. Type

The HF Name Data Store is a data store that is accessed by the HF Name Processor.

3.2.2.3. Purpose

The HF Name Data Store encodes the knowledge necessary for the HF Name Processor function of the LID to add information needed for the cultural identification of the input name.

3.2.2.4. Function

3.2.2.4.1. Table 3.2-2 contains a description of the data to be contained in the HF Name data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
NAME	character	24	*
NAMEFIELD	character	1	{G,S }
SCORE	integer	1	{1,2,3,4,5}
CULTURE	character	1	{H, A, O}

Table 3.2-2

3.2.2.4.2. The HF Name Processor uses the information provided in this data store when gathering detailed HF name cultural affinity information to be returned to LIA. High frequency given names and surnames for each of the three target cultural groups will be listed in the high frequency data store.

3.2.2.4.2.1. NAME indicates the literal string representation of the HF name.

3.2.2.4.2.2. NAMEFIELD indicates whether the score listed for the HF name applies to the Given Name (G) or to the Surname (S).

3.2.2.4.2.3. SCORE reflects the degree to which a name may be considered high frequency within the culture in question, and is the score assigned by the HF Name Processor when the HF name listed is found in the input name. For processing details, see section 3.1.4, High Frequency (HF) Name Processor Module Decomposition.

3.2.2.4.2.4. CULTURE indicates the cultural affinity with which the given NAME-NAMEFIELD-SCORE combination is associated.

3.2.2.4.2.5. A HF name string may appear in the HF Names Data Store multiple times if it is associated with multiple cultural affinities, or if it associated with a different frequency score in the given name and surname. In this instance, the correct score must be assigned for each CULTURE, NAMEFILED combination associated with the HF name in question.

3.2.2.4.2.5.1. For an example of scoring of an input name using raw scores returned by agents and

the LID Parameters, see Figure 3-4.

3.2.2.5. Subordinates

None.

3.2.3. TAQ Data Store Data Decomposition

3.2.3.1. Identification

This data store is referred to as the TAQ Data Store.

3.2.3.2. Type

The TAQ Data Store is a data store that is accessed by the TAQ Processor.

3.2.3.3. Purpose

The TAQ Data Store encodes the knowledge necessary for the TAQ Processor function of the LID to add information needed for the cultural identification of the input name.

3.2.3.4. Function

3.2.3.4.1. Table 3.2-3 contains a description of the data to be contained in the TAQ data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
TAQ	character	24	*
NAMEFIELD	character	1	{G,S,B}
SCORE	integer	10	{1,2,3,4,5}
CULTURE	integer	3	1..1,000

Table 3.2-3

3.2.3.4.2. The TAQ Processor uses the information provided in this data store when gathering detailed TAQ - cultural affinity information to be returned to LIA. TAQ values for each of the three target cultural groups will be listed in the TAQ data store.

3.2.3.4.2.1. TAQ indicates the literal string representation of the Title, Affix or Qualifier particle. Note that only free

Morphemes are included in the TAQ data store, so, by definition, all TAQs are implicitly bounded by white space.

3.2.3.4.2.2. NAMEFIELD indicates whether the score listed for the given TAQ particle applies to the Given Name (G), to the Surname (S), or to Both (B).

3.2.3.4.2.2.1. In the event that the NAMEFIELD is listed as "B", the associated TAQ is defined as "in position" whether it is found in the given name or in the surname field in the input name, and is scored accordingly.

3.2.3.4.2.3. SCORE is a score for the given TAQ-NAMEFIELD-CULTURE combination. The TAQ scores will reflect the predictive value of the TAQ particle for the culture with which it is associated. This is the score assigned by the TAQ Processor when the TAQ particle listed is found in the input name. For processing details, see section 3.1.5, Title, Affix, Qualifier (TAQ) Processor Module Decomposition.

3.2.3.4.2.4. CULTURE indicates the cultural affinity with which the given TAQ-NAMEFIELD-SCORE combination is associated.

3.2.3.4.2.5. A TAQ particle may appear in the TAQ Data Store multiple times if it is associated with multiple cultural affinities. In this instance, the correct score must be assigned for each cultural affinity associated with the TAQ value in question.

3.2.3.4.2.5.1. For an example of scoring of an input name using raw

scores returned by agents and the LID Parameters, see Figure 3-4.

3.2.3.5. Subordinates

None.

3.2.4. Morphological Data Store Data Decomposition

3.2.4.1. Identification

This data store is referred to as the Morphological Data Store.

3.2.4.2. Type

The Morphological Data Store is a data store that is accessed by the Morphological Processor.

3.2.4.3. Purpose

The Morphological Data Store encodes the knowledge necessary for the Morphological Processor function of the LID to intelligently process the input name, evaluating evidence based on culturally-specific Morphemes, and adding this to information needed for the cultural identification of the input name.

3.2.4.4. Function

3.2.4.4.1. Table 3.2-4 contains a description of the data to be contained in the Morphology data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
MORPHEME	character	24	*
NAMEFIELD	character	1	{G, S, B}
MORHTYPE	character	1	{E, P, S, I, A}
SCORE	integer	1	{1, 2, 3, 4, 5}
CULTURE	character	1	{A, H, O}

Table 3.2-4

3.2.4.4.2. The Morphological Processor uses the information provided in this data store when gathering detailed Morpheme - cultural affinity

information to be returned to LIA. Morpheme values for each of the three target cultural groups will be listed in the Morphological Data Store.

3.2.4.4.2.1. MORPHEME indicates the literal string representation of the Morpheme. Note that only bound Morphemes are included in the Morphological data store, so, by definition, all Morphemes are intended to be located as substrings of individual segments of the input name.

3.2.4.4.2.2. NAMEFIELD indicates whether the score listed for the given Morpheme applies to the Given Name (G), to the Surname (S), or to Both (B).

3.2.4.4.2.2.1. In the event that the NAMEFIELD is listed as "B", the associated MORPHEME is defined as "in position" whether it is found in the given name or in the surname field in the input name, and is scored accordingly.

3.2.4.4.2.3. MORPHTYPE indicates the linguistic distribution of the MORPHEME.

3.2.4.4.2.3.1. Prefixes (P) are substrings which begin in the first character position of a name segment.

3.2.4.4.2.3.2. INFIXES (I) are substrings which begin in a character position in the name segment which is not the first, and end in a character position in the name segment that is not the last. They are substrings that are neither at the beginning nor the end of the name segment.

3.2.4.4.2.3.3. SUFFIXES (S) are substrings which end in the final character position of a name segment.

3.2.4.4.2.3.4. A MORPHEME for which the MORPHTYPE is indicated as EDGE (E) can be found as either a PREFIX or a SUFFIX in a name segment in the input name.

3.2.4.4.2.3.5. A MORPHEME for which the MORPHTYPE is indicated as ALL (A) can be found anywhere in a name segment in the input name.

3.2.4.4.2.3.6. MORPHEMEs that are found in positions other than those indicated by the corresponding MORPHTYPE are not assigned any points for the purpose of identifying the cultural affinity of the input name.

3.2.4.4.2.4. SCORE is a score for the given MORPHEME-NAMEFIELD-MORPHTYPE-CULTURE combination. The MORPHEME scores will reflect the predictive value of the Morpheme for the culture with which it is associated. This is the score assigned by the Morphological Processor when the Morpheme listed is found in the input name. For processing details, see section 3.1.6, Morphological Processor Module Decomposition.

3.2.4.4.2.5. CULTURE indicates the cultural affinity with which the given MORPHEME-MORPHTYPE-

NAMEFIELD-SCORE combination is associated.

3.2.4.4.2.6. A Morpheme may appear in the Morphological Data Store multiple times if it is associated with multiple cultural affinities, or if it can be associated with multiple values of NAMEFIELD and/or MORPHTYPE for a given cultural affinity. In this instance, the correct score must be assigned for each MORPHEME-MORPHTYPE-NAMEFIELD-CULTURE combination associated with the Morpheme in question.

3.2.4.4.2.6.1. For an example of scoring of an input name using raw scores returned by agents and the LID Parameters, see Figure 3-4.

3.2.4.5. Subordinates

None.

3.2.5. Ngram Data Store Data Decomposition

3.2.5.1. Identification

This data store is referred to as the Ngram Data Store.

3.2.5.2. Type

The Ngram Data Store is a data store that is accessed by the Ngram Processor.

3.2.5.3. Purpose

The Ngram Data Store encodes the knowledge necessary for the Ngram Processor function of the LID to add evidence based on the distribution of culturally salient Ngrams to information needed for the cultural identification of the input name.

3.2.5.4. Function

3.2.5.4.1. Table 3.2-5 contains a description of the data to be contained in the Ngram data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
NGRAM	character	10	*
NAMEFIELD	character	1	{G, S, B}
NGRAMTYPE	character	1	{E, P, I, S, A}
SCORE	integer	1	{1, 2, 3, 4, 5}
CULTURE	character	1	{A, H, O}

Table 3.2-5

3.2.5.4.2. The Ngram Processor uses the information provided in this data store when gathering detailed cultural affinity information to be returned to LIA. Ngram values for each of the three target cultural groups will be listed in the Ngram Data Store.

3.2.5.4.2.1. NGRAM indicates the literal string representation of the Ngram. Note that all Ngrams are intended to be located as substrings of individual segments of the input name.

3.2.5.4.2.2. NAMEFIELD indicates whether the score listed for the given Ngram applies to the Given Name (G), to the Surname (S), or to Both (B).

3.2.5.4.2.2.1. In the event that the NAMEFIELD is listed as "B", the associated NGRAM is defined as "in position" whether it is found in the given name or in the surname field in the input name, and is scored accordingly.

3.2.5.4.2.3. NGRAMTYPE indicates the linguistic distribution of the NGRAM.

3.2.5.4.2.3.1. PREFIXES (P) are substrings which begin in the first character position of a name segment.

3.2.5.4.2.3.2. INFIXES (I) are substrings which begin in a character position in the name segment which is not the first, and end in a character position in the name segment that is not the last. They are substrings that are neither at the beginning nor the end of the name segment.

3.2.5.4.2.3.3. SUFFIXES (S) are substrings which end in the final character position of a name segment.

3.2.5.4.2.3.4. An NGRAM for which the NGRAMTYPE is indicated as EDGE (E) can be found as either a PREFIX or a SUFFIX in a name segment in the input name.

3.2.5.4.2.3.5. An NGRAM for which the NGRAMTYPE is indicated as ALL (A) can be found anywhere in a name segment in the input name.

3.2.5.4.2.3.6. NGRAMs that are found in positions other than those indicated by the corresponding NGRAMTYPE are not assigned any points for the purpose of identifying the cultural affinity of the input name.

3.2.5.4.2.4. SCORE is a score for the given NGRAM-NAMEFIELD-NGRAMTYPE-CULTURE combination. The NGRAM scores will reflect the predictive value of the Ngram for the culture with which it is

associated. This is the score assigned by the Ngram Processor when the given Ngram is found in the input name. For processing details, see section 3.1.7, Ngram Processor Module Decomposition.

3.2.5.4.2.5. CULTURE indicates the cultural affinity with which the given NGRAM-NGRAMTYPE-NAMEFIELD-SCORE combination is associated.

3.2.5.4.2.6. An Ngram may appear in the Ngram Data Store multiple times if it is associated with multiple cultural affinities, or if it can be associated with multiple values of NAMEFIELD and/or NGRAMTYPE for a given cultural affinity. In this instance, the correct score must be assigned for each NGRAM-NGRAMTYPE-NAMEFIELD-CULTURE combination associated with the Ngram in question.

3.2.5.4.2.6.1. For an example of scoring of an input name using raw scores returned by agents and the LID Parameters, see Figure 3-4.

3.2.5.5. Subordinates

None.

Factors:	In Field: 10 OutField: 8	In Field: 8 OutField: 6	In FieldSN: 5 OutFieldSN: 3 In FieldGN: 4 OutFieldGN: 2	In FieldSN: 3 OutFieldSN: 2 In FieldGN: 2 OutFieldGN: 1	In FieldSN: 5 OutFieldSN: 3 In FieldGN: 4 OutFieldGN: 2

High Frequency SN			
H	S	Garcia	3
H	S	Salazar	2
H	S	Sambrano	1
O	S	Greco	5
O	B	Giuliano	2
O	S	Silvestri	1

High Frequency GN			
H	G	Jose	3
H	G	Francisco	2
H	G	Mario	1
O	B	Luigi	3
O	G	Antonio	2
O	G	Adalberto	1

Prefixes			
H	S	de	1
H	S	la	1
H	S	las	1
O	B	il	1
O	B	el	1
O	B	lo	1

N-Grams			
H	S	-ndez	3
H	S	-guez	2
H	S	-illo	1
O	S	-illo	3
O	B	-ini	2
O	S	-agio	1

Morphology			
A	S	adin	3
A	S	eddin	2
A	S	uddin	1
O	S	etto	5
O	S	etti	4
O	S	ini	2

N.B.:The data shown here are for the purpose of illustration only and do not necessarily reflect actual table values.

Sample name: DELGADILLO DE GARCIA, JOSE ANTONIO

	-illo		Antonio	de		Garcia		Jose	
Hispanic	(3 * 1)	+		(5 * 1)	+	(10 * 3)	+	(8 * 3)	= 62
Arabic	*note that none of these elements are marked as Arabic in the sample data above.								= 0
Other	(3 * 3)	+	(8 * 2)						= 25

N.B. The data shown here are for the purposes of illustration only and are not intended to make any statement about actual table values or parameter settings.

Figure 3-4

3.2.6. Digraph Distribution Data Store Data Decomposition

3.2.6.1. Identification

This data store is referred to as the Digraph Data Store.

3.2.6.2. Type

The Digraph Data Store is a data store that is accessed by the Digraph Distribution Processor.

3.2.6.3. Purpose

The Digraph Data Store encodes the knowledge necessary regarding the statistical distribution of digraphs within a given culture. It is this information that drives the Digraph Distribution Processor.

3.2.6.4. Function

3.2.6.4.1. Table 3.2-6 contains a description of the data to be contained in the Digraph data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
DI	character	2	*
SCORE	long	3.4	{-50.0000 - +50.0000}
CULTURE	character	1	{A, H}

Table 3.2-6

3.2.6.4.2. The Digraph Processor uses the information provided in this data store when determining the contribution that the distribution of digraphs in the input name will have in determining the cultural affinity of that name Digraph Distribution statistics will be listed in the Digraph Data Store for each of the specific cultures being identified. That is, in the current implementation, Digraph Distribution statistics will be listed for Arabic and Hispanic, but not for "Other".

3.2.6.4.2.1.1. DI indicates the literal string representation of the digraph. Note that digraphs may include all alphabetical characters as well as the word-boundary character "#".

3.2.6.4.2.2. SCORE reflects the predictive value of the digraph for the culture with which it is associated. This is the score used by the Digraph Distribution Processor when the given digraph is found in the input name. For processing details, see section 3.1.9, Digraph Distribution Processor Module Decomposition.

3.2.6.4.2.3. CULTURE indicates the cultural affinity with which the given DI-SCORE combination is associated.

3.2.6.5. Subordinates

None.

3.2.7. Trigraph Distribution Data Store Data Decomposition

3.2.7.1. Identification

This data store is referred to as the Trigraph Data Store.

3.2.7.2. Type

The Trigraph Data Store is a data store that is accessed by the Digraph Distribution Processor. The Digraph Distribution Processor takes initial and final trigraphs into account in producing a digraph distribution score for the input name.

3.2.7.3. Purpose

The Trigraph Data Store encodes the knowledge necessary regarding the statistical distribution of trigraphs within a given culture. This information is taken into account in the Digraph Distribution Processor, since name boundaries tend to be highly indicative of the cultural affinity of the name.

3.2.7.4. Function

3.2.7.4.1. Table 3.2-7 contains a description of the data to be contained in the Trigraph data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
TRI	character	3	*
SCORE	long	3.4	{-50.0000 - +50.0000}
CULTURE	character	1	{A, H}

Table 3.2-7

3.2.7.4.2. The Digraph Processor uses the information provided in this data store when determining the contribution that the distribution of initial and final trigraphs in the input name will have in determining the cultural affinity of that name Trigraph Distribution statistics will be listed in the Trigraph Data Store for each of the specific cultures being identified. That is, in the current implementation, Trigraph Distribution statistics will be listed for Arabic and Hispanic, but not for "Other".

3.2.7.4.2.1. DI indicates the literal string representation of the trigraph. Note that trigraphs may include all alphabetical characters as well as the word-boundary character "#".

3.2.7.4.2.2. SCORE reflects the predictive value of the trigraph for the culture with which it is associated. This is the score used by the Digraph Distribution Processor when the given trigraph is found in the input name. For processing details, see section 3.1.9, Digraph Distribution Processor Module Decomposition.

3.2.7.4.2.3. CULTURE indicates the cultural affinity with which the given DI-SCORE combination is associated.

3.2.7.4.3. Trigraph Distribution statistics for only initial and final trigraphs will be included in the Trigraph Data Store.

3.2.7.5. Subordinates

None.

3.2.8. Digraph Distribution Processor Parameter Data Store Data Decomposition

3.2.8.1. Identification

This data store is referred to as the Digraph Processor Parameter Data Store.

3.2.8.2. Type

The Digraph Processor Parameter Data Store is a data store that is accessed by the Digraph Distribution Processor.

3.2.8.3. Purpose

The Digraph Processor Parameter Data Store contains adjustments that must be made to the digraph distribution scores computed by the Digraph Distribution Processor due to the fact that some cultures are over-represented in the digraph model.

3.2.8.4. Function

3.2.8.4.1. Table 3.2-8 contains a description of the data to be contained in the Trigraph data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
SKEW	integer	3	{-999 - +999}
CULTURE	character	1	{A, H}

Table 3.2-8

3.2.8.4.2. The Digraph Processor uses the information provided in this data store when determining the final digraph distribution score to assign to the input name. A SKEW will be specified in the Digraph Processor Parameter Data Store for each of the specific cultures being identified. That is, in the current implementation, a SKEW will be listed for Arabic and Hispanic, but not for "Other".

3.2.8.4.2.1. SKEW indicates the value to be added to or subtracted from the raw digraph distribution score by the digraph distribution processor to level data distribution differences.

3.2.8.4.2.2. CULTURE indicates the cultural affinity with which the given SKEW is associated.

3.2.8.5. Subordinates

None.

3.2.9. Threshold Parameter Data Store Data Decomposition

3.2.9.1. Identification

This data store is referred to as the Threshold Parameter Data Store.

3.2.9.2. Type

The Threshold Parameter Data Store is a data store that is accessed by the Intermediate Decision Processor 1 (IDP1),

the Intermediate Decision Processor 2 (IDP2), and the Final Decision Processor.

3.2.9.3. Purpose

The Threshold Parameter Data Store contains information regarding thresholds that must be met in order for the input name to be identified as belonging to a particular target culture.

3.2.9.4. Function

3.2.9.4.1. Table 3.2-9 contains a description of the data to be contained in the Threshold Parameter data store.

DATA NAME	DATA TYPE	DATA WIDTH	POSSIBLE VALUES
CULTURE	character	1	{A, H, O}
LID_THRESHOLD	integer	3	{0 - 999}
DI_THRESHOLD	float	3.4	{-999.9999 - +999.9999}
UNDER_LID_ THRESHOLD	integer	3	{0 - 999}
UNDER_DI_ THRESHOLD	integer	3	{0 - 999}

Table 3.2-9

3.2.9.4.2. The three "decision processor" modules (IDP1, IDP2, and the Final Decision Processor) use the information provided in this data store when determining whether enough information has been accumulated to identify the input name as belonging to a particular culture. LID_THRESHOLD and UNDER_LID_THRESHOLD data values will be specified in the Threshold Parameter Data Store for each of the cultures being identified, including "Other". DI_THRESHOLD and UNDER_DI_THRESHOLD values will be specified for specific cultures only (i.e. Hispanic and Arabic).

3.2.9.4.2.1. CULTURE indicates the cultural affinity with which the given threshold is associated.

3.2.9.4.2.2. LID_THRESHOLD is used by IDP1 in determining whether enough information has been accumulated to identify the input name as belonging to a particular culture. For processing information, see section 3.1.8, Intermediate Decision Processor 1 (LID Decision) Module Decomposition.

3.2.9.4.2.3. DI_THRESHOLD is used in IDP2 in determining whether enough information has been accumulated to identify the input name as belonging to a particular culture. For processing information, see section 3.1.11, Intermediate Decision Processor 2 (Digraph Decision) Module Decomposition.

3.2.9.4.2.4. UNDER_LID_THRESHOLD is used by the Final Decision Processor, and indicates the amount by which a name can fall short of the LID_THRESHOLD and still be considered for membership in a particular culture, provided that other criteria are met. As such, UNDER_LID_THRESHOLD defines a range of values (between the UNDER_LID_THRESHOLD and the LID_THRESHOLD) that, when considered in conjunction with other evidence, can result in the input name's being identified as belonging to the culture in question. For processing information see section 3.1.12, Final Decision Processor Module Decomposition and Figure 3-2.

3.2.9.4.2.5. UNDER_DI_THRESHOLD is used by the Final Decision Processor, and indicates the amount by which a name can fall short of the DI_THRESHOLD and still be considered for membership in a particular culture, provided that other criteria are met. As such, UNDER_DI_THRESHOLD defines a range of values (between the UNDER_DI_THRESHOLD and the DI_THRESHOLD) that, when considered in conjunction with other evidence, can result in the input name's being identified as belonging to the culture in question. For processing information, see section 3.1.12, Final Decision Processor Module Decomposition and Figure 3-2.

3.2.9.5. Subordinates

None.

3.2.10. COB Proximity (COBPROX) Data Store Data Decomposition

3.2.10.1. Identification

This data store is referred to as the COBPROX Data Store.

3.2.10.2. Type

The COBPROX Data Store is a data store that is accessed by the Final Decision Processor.

3.2.10.3. Purpose

The COBPROX Data Store contains information enabling the Final Decision Processor to determine which COBs are to be considered as related when determining the cultural affinity of the input name. For processing information, see section 3.1.12.4.5 and Figure 3-2.

3.2.10.4. Function

3.2.10.4.1.1. ANC-E will use the CLASS-E COBPROX Data Store ("partition table") to fill this function.

3.2.10.5. Subordinates

None.

APPENDIX A: DETAILED EXAMPLE OF ANC-E LID PROCESSING

LIA Data Stores

High Frequency			
H	S	Garcia	3
H	S	Salazar	2
H	S	Sambrano	1
A	B	Mahmoud	4
A	B	Jaffar	2
O	S	Silvestri	1

N-Grams				
H	S	S	ndez	3
H	S	G	far	1
A	B	P	bous	1
A	B	S	fia	2
O	S	E	agio	1
O	S	I	ahmo	5

Morphology				
A	B	S	addin	2
A	B	S	edin	3
A	B	S	uddin	3
O	S	P	etto	1
O	S	S	etti	2
O	S	S	ini	1

TAO			
H	S	de	1
H	S	la	1
H	S	las	1
A	B	bin	3
O	B	el	1
O	B	lo	1

Digraph Distribution Processor Data Stores

Digraphs		
A	ez	-1.0422
A	nt	22.8733
A	bd	38.7221
H	bd	1.0572
H	ez	42.5947
H	ri	16.1242

Trigraphs		
A	ez#	-10.0422
A	#nt	-48.1743
A	bd#	48.4551
H	bd#	-32.1742
H	ez#	47.5327
H	#ri	11.1242

Digraph Processor Parameters	
A	44.2331
H	-32.8765

LID Parameters			
HFNAME	S	10	8
HFNAME	G	8	6
TAO	S	5	3
TAO	G	4	2
MORPHOLOGY	S	5	3
MORPHOLOGY	G	4	2
NGRAM	S	4	3
NGRAM	G	3	2

Threshold Parameters				
A	50	1.4532	10	8
H	73	20.5000	5	6
O	38	(null)	100	3

COBPROX

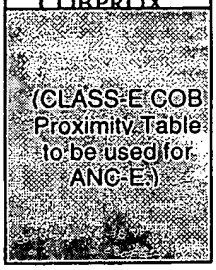


Fig. 5

N.B.:The data shown here are for the purpose of illustration only and do not necessarily reflect actual values

Sample ANC-E Data Stores

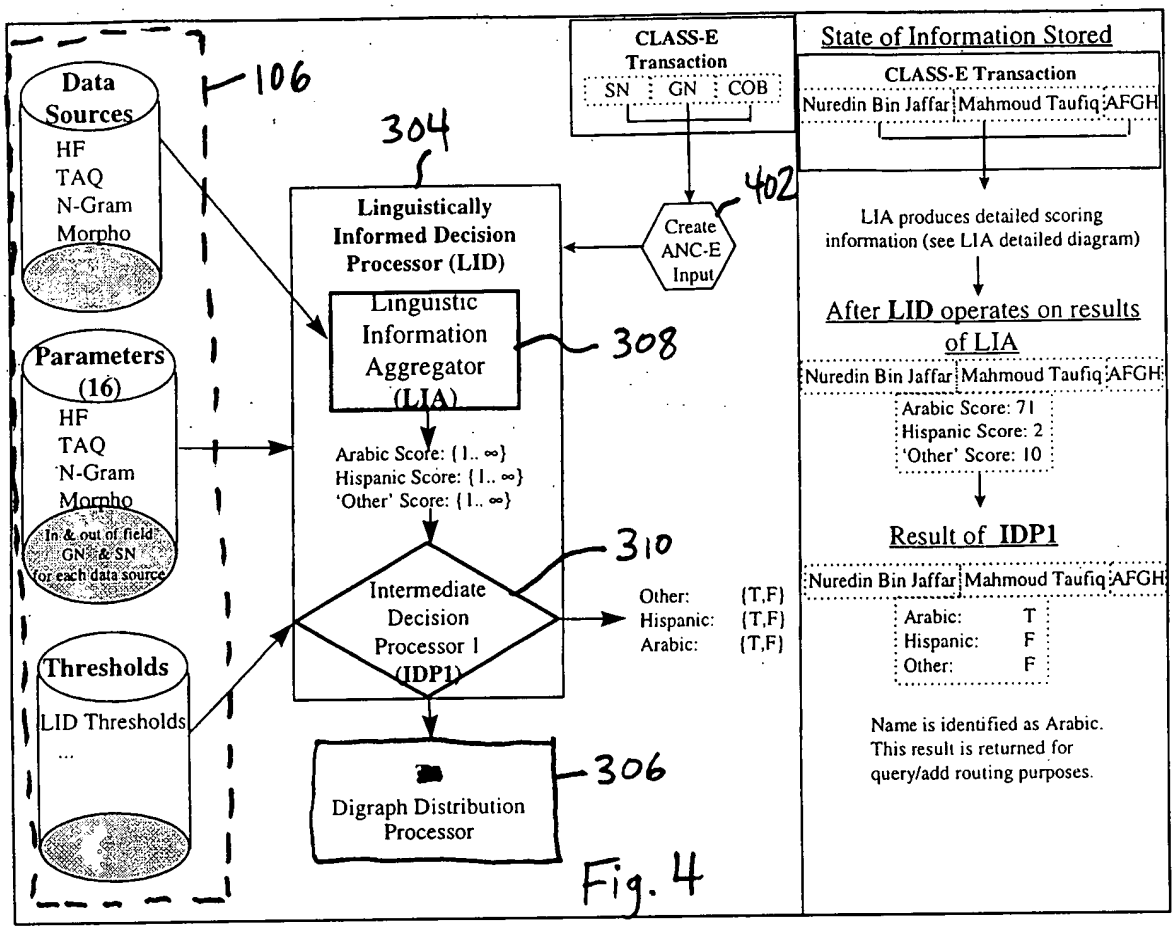
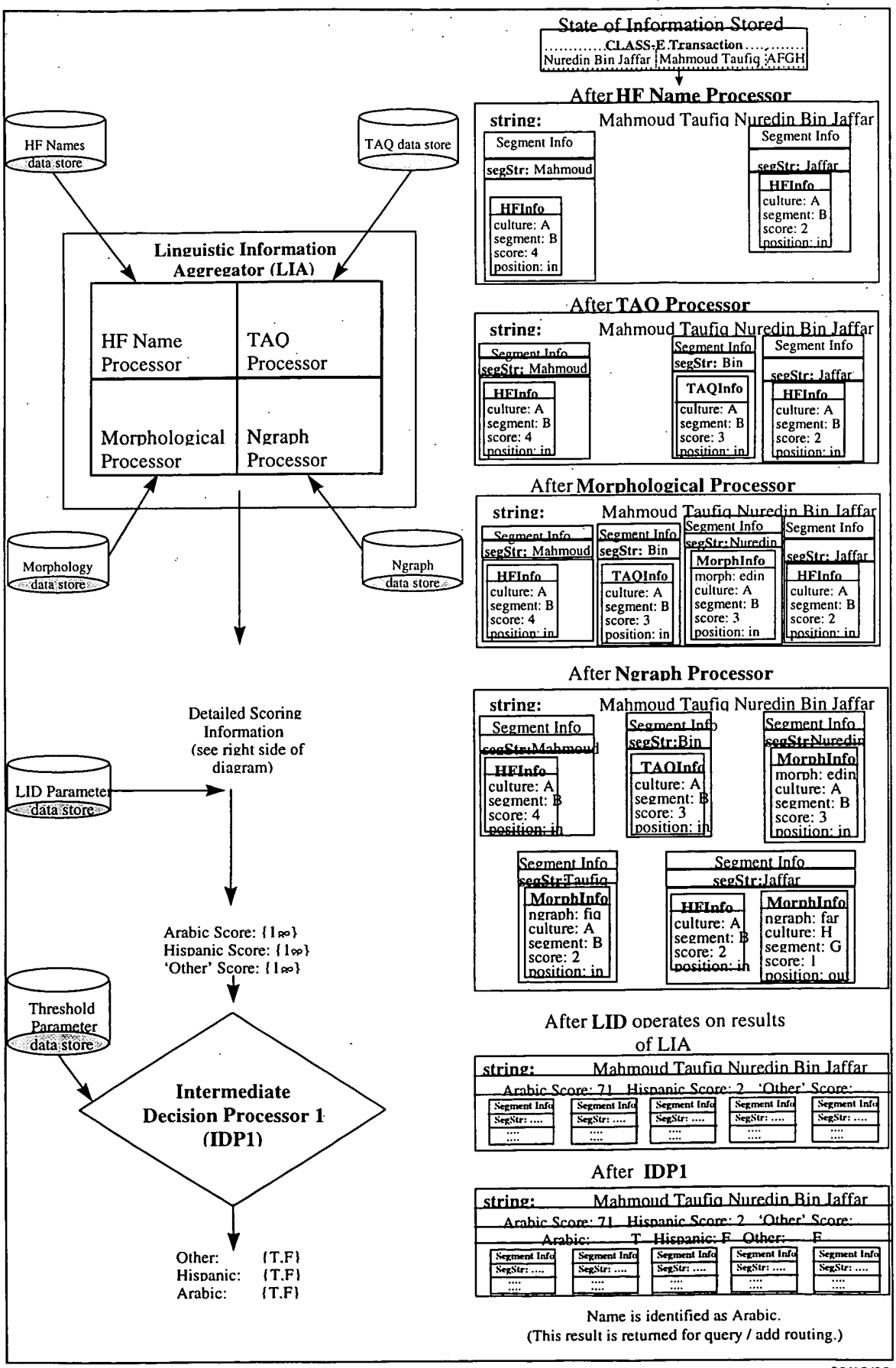


Fig. 4

Overview of LID Processing

Detailed View of LID Processing
(p. 60)


State of Information Stored
CLASS-E.Transaction
Nuredin Bin Jaffar | Mahmoud Taufiq | AFGH
After HF Name Processor
string: Mahmoud Taufiq Nuredin Bin Jaffar
Segment Info
segStr: Mahmoud
HFInfo
culture: A
segment: B
score: 4
position: in
Segment Info
segStr: Jaffar
HFInfo
culture: A
segment: B
score: 2
position: in
HF Names data store
TAQ data store
Linguistic Information Aggregator (LIA)
HF Name Processor
TAQ Processor
Morphological Processor
Ngraph Processor
Morphology data store
Ngraph data store
After TAQ Processor
string: Mahmoud Taufiq Nuredin Bin Jaffar
Segment Info
segStr: Mahmoud
HFInfo
culture: A
segment: B
score: 4
position: in
Segment Info
segStr: Bin
TAQInfo
culture: A
segment: B
score: 3
position: in
Segment Info
segStr: Jaffar
HFInfo
culture: A
segment: B
score: 2
position: in
After Morphological Processor
string: Mahmoud Taufiq Nuredin Bin Jaffar
Segment Info
segStr: Mahmoud
HFInfo
culture: A
segment: B
score: 4
position: in
Segment Info
segStr: Bin
TAQInfo
culture: A
segment: B
score: 3
position: in
Segment Info
segStr: Nuredin
MorphInfo
morph: edin
culture: A
segment: B
score: 3
position: in
Segment Info
segStr: Jaffar
HFInfo
culture: A
segment: B
score: 2
position: in
After Ngraph Processor
string: Mahmoud Taufiq Nuredin Bin Jaffar
Segment Info
segStr:Mahmoud
HFInfo
culture: A
segment: B
score: 4
position: in
Segment Info
segStr:Bin
TAQInfo
culture: A
segment: B
score: 3
position: in
Segment Info
segStr:Nuredin
MorphInfo
morph: edin
culture: A
segment: B
score: 3
position: in
Segment Info
segStr:Taufiq
MorphInfo
ngraph: fiq
culture: A
segment: B
score: 2
position: in
Segment Info
segStr:Jaffar
HFInfo
culture: A
segment: B
score: 2
position: in
MorphInfo
ngraph: far
culture: H
segment: G
score: 1
position: out
Detailed Scoring Information (see right side of diagram)
LID Parameter data store
Arabic Score: {1∞}
Hispanic Score: {1∞}
'Other' Score: {1∞}
Threshold Parameter data store
Intermediate Decision Processor 1 (IDP1)
Other: {T,F}
Hispanic: {T,F}
Arabic: {T,F}
After LID operates on results of LIA
string: Mahmoud Taufiq Nuredin Bin Jaffar
Arabic Score: 71 Hispanic Score: 2 'Other' Score:
Segment Info
SegStr: ....
After IDP1
string: Mahmoud Taufiq Nuredin Bin Jaffar
Arabic Score: 71 Hispanic Score: 2 'Other' Score:
Arabic: T Hispanic: F Other: F
Segment Info
SegStr: ....
Name is identified as Arabic.
(This result is returned for query / add routing.)